Whole-Body End-Pose Planning for Legged Robots on Inclined Support Surfaces in Complex Environments

Whole-Body End-Pose Planning for Legged Robots on Inclined Support Surfaces in Complex Environments

Henrique Ferrolho, Wolfgang Merkt, Yiming Yang, Vladimir Ivan, and Sethu Vijayakumar

***Abstract*— Planning balanced whole-body reaching configurations is a fundamental problem in humanoid robotics on which manipulation and locomotion planners depend on. While finding valid whole-body configurations in free space and on flat terrains is relatively straightforward, the problem becomes extremely challenging when obstacle avoidance is taken into account, and when balancing on more complex terrains, such as inclined supports or steps. Previous work using *Paired Forward-Inverse Dynamic Reachability Maps* demonstrated fast end-pose planning on flat terrains at different heights by decomposing the kinematic structure and leveraging combinatorics. In this paper, we present an efficient whole-body end-pose planning framework capable of finding collision-free whole-body configurations in complex environments and on sloped support regions. The main contributions in this paper are twofold: (i) the integration of contact property information of support regions into both precomputation and online planning stages, including whole-body static equilibrium robustness, and (ii) the proposal of a more informed and meaningful sampling strategy for the lower-body. We focus on humanoid robots throughout the paper, but all the principles can be applied to legged platforms other than bipedal robots. We demonstrate our method on the NASA Valkyrie humanoid platform with 38 Degrees of Freedom (DoF) over inclined supports. Analysis of the results indicate both higher success rates — greater than 95% and 80% on obstacle-free and highly cluttered environments, respectively — and shorter computation times compared to previous methods.**

I. Introduction

Humanoid robots are complex systems designed to perform dexterous tasks in environments designed and engineered for people (cf. Figure 1). While their key advantage is the ability to operate in uneven terrain and unstructured environments such as disaster sites or outdoor environments, they require active control to maintain balance, thus rendering fast planning and control challenging problems.

Directly planning motion which includes locomotion and manipulation in a single formulation is non-trivial as switching contacts and balance have to be taken into account, and available contact-implicit trajectory optimization formulations can easily take hours to compute. Thus, it is a common approach to decompose the overall planning into subproblems [1], [2], e.g., to first plan a pre-manipulation stance and configuration (*end-pose planning* [2]) and use this as a goal for footstep [3] or acyclic contact planners [1] to generate a guide trajectory along with a sequence of contacts that navigates to the pre-manipulation stance. The final configuration further serves as an input to a whole-body motion planner [4] to compute manipulation tasks with the feet assumed to be stationary.

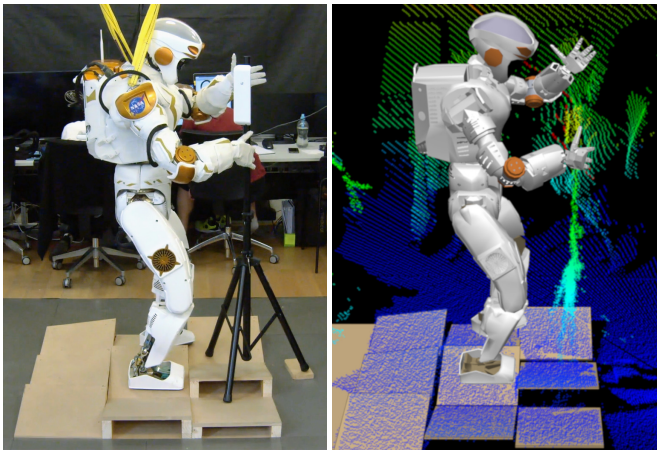


Fig. 1: End-pose planning for a bimanual manipulation task where the robot has to reach for an antenna in a complex environment of sloped support surfaces. Left: photo of the robot in a pre-grasp stance. Right: visualization of the perceived 3D point cloud, fitted terrain model, and the planned whole-body configuration. Our method automatically adapts to the constraints imposed by the environment and chooses collision-free statically-balanced stance locations and whole-body configurations.

As such, with a focus on manipulation in complex environments, the success of these planning pipelines hinges on the quality of the final whole-body configuration. However, finding an appropriate pre-manipulation stance and configuration is not trivial due to the necessity of considering collision avoidance in close proximity, contact support properties, and robot manipulability. Therefore, traditionally, a pre-grasp whole-body configuration was either provided by a human operator or based on inverse kinematics without collision avoidance, requiring the operator to manually confirm validity [5]. This often resulted in little exploration of the redundancy of high-DoF platforms and did not leverage repositionability, making it unsuitable for complex environments. Furthermore, such human-in-the-loop processes become a limiting factor for autonomous operation.

Direct optimization-based formulations in this setting are unlikely to succeed: the problem is highly discontinuous and non-convex, with many local minima. Thus, a good initialization seed is required, especially since many constraints and objectives are expensive to compute, do not provide gradient information, or are difficult to replace with proxy constraints.

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC, grant reference EP/L016834/1) and EU H2020 project Memory of Motion (MEMMO, project ID: 780684). The work has been performed at the University of Edinburgh under the Centre for Doctoral Training in Robotics and Autonomous Systems program.

All authors are with the Institute for Perception, Action, and Behaviour, School of Informatics, The University of Edinburgh (Informatics Forum, 10 Crichton Street, Edinburgh, EH8 9AB, United Kingdom). Email: henrique.ferrolho@ed.ac.uk.

TABLE I: Comparison of reachability-based end-pose planning methods.

Method	Task constraint	Feet placement	Assumptions
IRM [6], iDRM [2]	Single hand (1×6D)	x, y, yaw for mid-feet (1×3D)	No slip; Horizontal support surface; Feet with constant displacement and zero yaw between each other.
Paired Forward--Inverse DRM [7]	Bimanual (2×6D)	x, y, z, yaw for each foot (2×4D)	No slip; Horizontal support surface ($roll = pitch = 0$).
Our method	Bimanual (2×6D)	$x, y, z, roll, pitch, yaw$ for each foot (2×6D)	No slip.

In order to exploit the redundancy of humanoids, prior work has focused on storing valid, balanced configurations along with an encoding of manipulability and reachability information during an offline preprocessing step [6], [2], [7]. However, these works are limited to flat terrains by using simple stability criteria. Furthermore, for work leveraging kinematic splits and combinatorics, no consideration is taken to ensure that recombined samples are valid and satisfy constraints, increasing requirements for online checking and planning times. Finally, the questions of required dataset sizes and good sampling strategies are not addressed.

To this end, we extend the prior work by taking into account support contact properties during both the offline preprocessing and online planning stages to enable whole-body, bimanual end-pose planning on sloped surfaces. We estimate the static equilibrium robustness of a whole-body configuration during the preprocessing stage through an informed approximation of the upper-body to ensure that recombined samples provide good initialization seeds. We further extensively evaluate different dataset sizes and sampling criteria settings in a complex, random benchmark. Finally, we embed our algorithm in a planning pipeline with multiple failure recovery mechanisms ensuring that a valid pre-grasp configuration will be found if it exists within the dataset. With these contributions, our reachability encoding enables us to exploit the null space of the robot to efficiently compute collision-free whole-body configurations in cluttered environments. A comparison with related end-pose planning methods is shown in Table I.

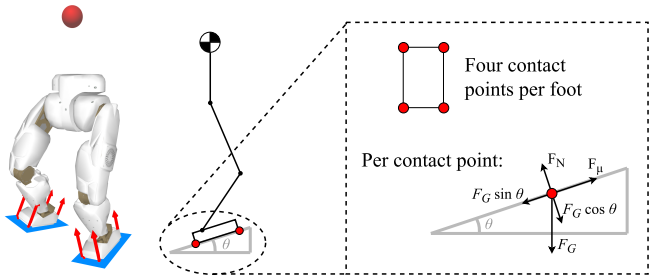
The planning framework has been validated in full-physics simulation with a model of the NASA Valkyrie humanoid robot with 38-DoF, demonstrating that the proposed method is able to find feasible pre-grasp whole-body configurations in complex environments with inclined support regions. We have further carried out hardware validation experiments of the simulated scenarios. An accompanying video is available at `https://youtu.be/tt6oYKuPI_A`.

II. Related Work

As a robotic system cannot reach every part of its workspace equally well, research has focused on characterizing the manipulability of workspace areas to find the best floating base placement by precomputing a map of the reachable workspace. The Reachability Map (RM) [8] records the reachable workspace regions of a fixed-base robot, allowing efficient query of whether a pose is reachable by the end-effector. Similarly, the Inverse Reachability Map (IRM) [9] encodes feasible stances for a mobile robot given an end-effector pose. Taking into account constraints such as stability and kinematic loop closure, Burget and Bennewitz [6] extended the IRM to humanoids. They used a dense coverage of the sampling space for a single-arm reaching task through deterministic sampling in configuration space ($\mathcal{C}$-space) and used the results directly without post-processing, allowing violation of reach constraints according to coverage/quality of samples and requiring online collision checking. However, these methods only store the kinematic reachability with collision checking performed online, significantly contributing to long planning times.

Yang et al. [2] introduced the inverse Dynamic Reachability Map (iDRM), which addresses this limitation by computing a custom mapping between $\mathcal{C}$-space and occupied workspace, therefore offloading collision checking to an offline preprocessing stage. This mapping was further used to initialize a reduced non-linear optimization problem. In order to achieve good dataset coverage, the authors used deterministic sampling in the workspace. Notwithstanding, iDRM relies on two limiting assumptions: (i) terrain of planning scenarios solely consists of a horizontal support surface; and (ii) the stance configuration always has the feet set parallel to each other and with fixed separation. These assumptions allow the explicit encoding of robust balance by only storing whole-body configurations that are in static equilibrium, i.e., configurations whose Center of Mass (CoM) projection falls within their support polygon, akin to [6]. A key limitation, however, is the trade-off between problem complexity and memory available to store enough samples in order to densely cover the $\mathcal{C}$-space.

More recently, Yang et al. [7] presented a novel end-pose planning algorithm which allows covering a larger or less constrained $\mathcal{C}$-space without exhausting available memory by decomposing the kinematic structure at the pelvis link and adding a recombination of upper- and lower-body samples valid for the selected environments. The authors explored both deterministic and uniform sampling techniques. The kinematic split allows leveraging the strengths of both forward and inverse dynamic reachability maps and creates a combinatorial pool of candidate whole-body configurations. The evaluation presented in [7] provided enough evidence to support that splitting the kinematic structure at the pelvis link is the most practical approach, considering the trade-off between coverage, planning success rate, and algorithm



(a) (b)

Fig. 2: (a) Example of a feasible lower-body configuration. The red sphere above the pelvis represents an approximation of the upper-body as a lumped point mass. The red arrow markers represent the contact normals for each contact point between the feet and their virtual support region (represented in blue). (b) Diagram showing the forces involved with contact on inclined surfaces: the gravitational force decomposes into an orthogonal and a tangential forces. Slippage occurs when the tangential component exceeds the frictional force $F_\mu = \mu F_N$, where μ is the friction coefficient between the foot sole and the contact surface materials.

runtime. The combinatorics of the modular maps hereby increased $\mathcal{C}$-space coverage, thus enabling the algorithm to compute pre-grasp whole-body configurations for problems requiring the feet to be placed on horizontal supports at different heights, as well as to freely place the feet relatively to each other. The latter is particularly useful for reaching tasks in environments that include small obstacles, e.g., boxes below waist height, resulting in expressive poses using the redundancy and flexibility of legged platforms, such as lunges and support steps. Nevertheless, this approach is limited to horizontal supports at different heights.

A key criterion for keeping underactuated systems balanced, especially on inclined terrains, is the robustness of the static equilibrium of contact forces. In assessing the static equilibrium of whole-body configurations, a key distinction is whether the support surfaces are flat or sloped, as the gravitational force decomposes into an orthogonal and a tangential component for non-zero inclinations (cf. Figure 2). For flat ground ($\theta = 0°$), there is no tangential component and a system is said to be in static equilibrium if the vertical projection of its CoM lies within the convex hull of the support polygon. To account for state estimation and modeling errors (e.g., elasticity in the legs) and robustness to small disturbances, a common approach is to shrink the contact polygon, thus creating the so-called support polygon.

For non-flat terrain, the CoM projection must lie within a non-linear convex set defined by the properties of each contact limb placement [10]. Different measures have been proposed in order to compare the robustness of the static equilibrium under arbitrary contacts. Caron et al. [11] proposed the capacity of a system to generate CoM accelerations within a polytope in the axial plane without a change in angular momentum. Barthélemy and Bidaud [12] proposed a robustness measure based on the radius of the largest hypersphere centered at the Gravito-Inertial Wrench (GIW) and fully contained inside the GIW cone. Subsequently, Del Prete et al. [13] proposed to account for robustness to errors in the contact-force tracking, i.e., to prevent the forces necessary to maintain equilibrium from being too close to the boundaries of the friction cones. The authors proved that the CoM-projection method extends to *quasi-flat* terrains but comes with a larger number of false negatives as, for example, the height difference between the contact points increases. Additionally, they propose an algorithm that outperforms (in terms of computation time) previous approaches by approximating friction cones with polytopes defined by a set of linear inequalities.

A robustness metric is essential to assess stability on uneven terrains. We will now describe how such a metric can be integrated with the reachability map at creation time to filter out the unstable poses and to reduce the map size.

III. Dynamic Reachability Maps Construction

A reachability map encodes the reachable poses of a robot link with respect to a given frame. The forward and inverse reachability maps, i.e., RM and IRM, thus only differ in the frame by which they are defined: the former in base frame, and the latter in end-effector frame. In simple terms, the forward RM encodes *how well a robot can reach different regions of its workspace*, whereas the IRM encodes *where the robot base can be placed, given a grasping target*. The major distinction between RM / IRM and their *dynamic* versions is the additional mapping information concerning *occupation* and *reach* lists. Both the Dynamic Reachability Map (DRM) and iDRM involve an initial step where the workspace is discretized into a bounded 3D voxel grid, $\mathbb{V}$. The resolution at which discretization is applied depends on the application's final purpose, and ultimately, represents a trade-off between memory usage, computation times, and planning accuracy.

Similarly to previous work, henceforth we will designate each map's reference frame link as *root link*, i.e., the base link for DRM, and one of the end-effector links for the iDRM. We will designate the mapped frames as *tip links*, i.e., the end-effector links (feet) for the lower-body DRM, and the base link and the remaining end-effector link for iDRM.

We discretize the reachable workspace during the offline preprocessing stage and we create two distinct lists for each voxel $v \in \mathbb{V}$: the *occupation list*, O_v, which maps to samples intersecting with the voxel v, and the *reach list*, R_v, containing the indices of the samples for which one of the *tip links* falls within the voxel v. These lists are then used online to efficiently invalidate samples that are in collision or cannot reach the target. For further detail, please refer to [7].

A. Upper-Body iDRM

The same sampling process for the upper-body iDRM from [7] is used. There exist two distinct variants of upper-body datasets: constrained and unconstrained. The unconstrained variant samples robot configurations within the full scope of the $\mathcal{C}$-space — this includes upper-body configurations where, in the specific case of a humanoid, the arms reach behind the robot. The constrained variant uses rejection sampling to discard samples whose tip links are not comfortably reaching a bounded region of space in front of the robot — the rationale

behind this being favoring the front side of the robot for increased manipulability where most sensor data is captured.

B. Robust Lower-Body Samples

A limiting assumption in our previous work [7] was that support regions would always be flat, even though they could be positioned at different heights, e.g., steps. As such, the lower-body dataset in that work consisted solely of configurations in which both feet were constrained to be horizontal (i.e., $roll = pitch = 0$). Nonetheless, in order to eliminate the need for that assumption, a lower-body dataset comprising non-horizontal feet is required.

A robustness measure for the equilibrium of a specified CoM position can be computed as proposed in [13] by solving the following Linear Program (LP):

$$\begin{aligned} \text{find} \quad & \mathbf{b}, b_0 \\ \text{maximize} \quad & b_0 \\ \text{subject to} \quad & G\mathbf{b} = D\mathbf{c} + \mathbf{d}, \\ & \mathbf{b} - \mathbb{1}b_0 \geq \mathbf{0}, \end{aligned} \tag{1}$$

where $\mathbf{b}$ is a vector of coefficients of the contact force generators ($f = G\mathbf{b}$), $b_0 \in \mathbb{R}$ is a scalar parameter proportional to the robustness measure, $\mathbf{c} \in \mathbb{R}^3$ is the CoM position, G is the matrix whose columns are the Gravito-Inertial Wrench (GIW) generators, D is the matrix mapping the CoM position to GIW, and $\mathbf{d}$ is the 6D vector containing the gravity component of the GIW. These variables can be computed from kinematic and dynamic properties of the robot model.

Since the kinematic structure has been split into upper- and lower-body parts, a whole-body configuration is not available during preprocessing, and consequently, neither a CoM position — albeit the masses of the platform's parts are known. In order to circumvent this problem, a lumped point mass can be added above the pelvis level to the lower-body model of the robot to approximate the upper-body CoM position (cf. Figure 2a). The position of the lumped point mass can be approximated by averaging the CoM positions of all upper-body samples contained in an upper-body dataset created beforehand:[1]

$$\mathbf{p}_{\text{lumped mass}} = \frac{1}{n} \sum_{i=1}^{n} f_c(\Upsilon_i) \tag{2}$$

where $\mathbf{p}$ is the approximated position of the lumped point mass being calculated, Υ is an upper-body dataset with n samples, and f_c is a function which returns the CoM position of a dataset entry Υ_i. At last, after this addition, a whole-body CoM position can be estimated, in turn allowing the calculation of an approximated robustness of a whole-body configuration static equilibrium. Despite such setup being only an approximation, it indeed provides a close estimation of the likely CoM for the whole-body robustness.

It is important to note that the equilibrium approximation depends on the direction of the gravity vector with respect to the floating base. Thus, in order to reuse the approximation computed offline during online recombination, the *roll* and *pitch* components of the pelvis are set to zero while maintaining the *yaw* component. This reduces the uncertainty and mismatch of the actual (vs. approximated) robustness measure when recombining individual upper- and lower-body samples to a whole-body candidate configuration.

C. Lower-Body DRM

The lower-body sampling process described in [7], similarly to [6], is carried out deterministically by stepping through joint range using fixed increments while constraining the feet to the flat surface. Instead, we use a pseudo-random rejection sampling procedure coupled with the static equilibrium robustness measure described in III-B to generate the datasets analyzed in this study. A sampled lower-body configuration is admitted for storage in the dataset being generated if and only if its static equilibrium robustness is greater than or equal to a certain threshold, R_{min}. This is in order to assure that (a) the forces necessary to maintain the equilibrium of the stored samples are not too close to the boundaries of the linear approximation of the friction cones, and (b) the amount of torque each joint is allowed to exert in order to achieve the pose is limited. We explored different R_{min} values during our evaluation and present the results in Table IV.

Additionally, the sampling strategy we suggest also considers the distance between feet before admitting a configuration to be stored in a dataset. This is due to the walking controller employed in the task-planning framework we use: limiting the maximum distance between the feet ensures that the whole-body configurations returned by the end-pose planner can be transitioned into. Since the method we present in this paper can be generalized to any legged platform, even beyond bipedal systems, this decision variable is left out to be dictated by the targeted robot platform.[2]

IV. End-Pose Planning on Inclined Terrain

Figure 3 highlights the pipeline we are proposing for whole-body end-pose planning on inclined terrains. Two essential prerequisites for the pipeline to function are two datasets generated offline, storing upper- and lower-body samples. Their computation has been addressed in III-A and III-C.

A planning request is triggered when a problem description is fed into the pipeline. This includes a bimanual grasping target and the environment information — including scene obstacles and support regions. Firstly, in *Stage 1*, we calculate a set of candidate whole-body configurations that are collision-free and satisfy the task constraints. This calculation is performed by the Paired Forward-Inverse DRM module (see [7] for more details). After the set is complete, the candidates are sorted according to the following cost function:

$$f(\mathbf{q}) = w_T ||T_{rhand}(\mathbf{q}^*) - T_{rhand}(\mathbf{q})|| + \frac{w_R}{R(\mathbf{q})}, \tag{3}$$

[1] We used the Υ_{4M_c} upper-body dataset presented in Table II to compute an approximation of the lumped point mass position.

[2] For the purpose of this study, we have opted to use a threshold of $0.5\,\text{m}$ for the maximum x-y distance between feet of the NASA Valkyrie humanoid robot.

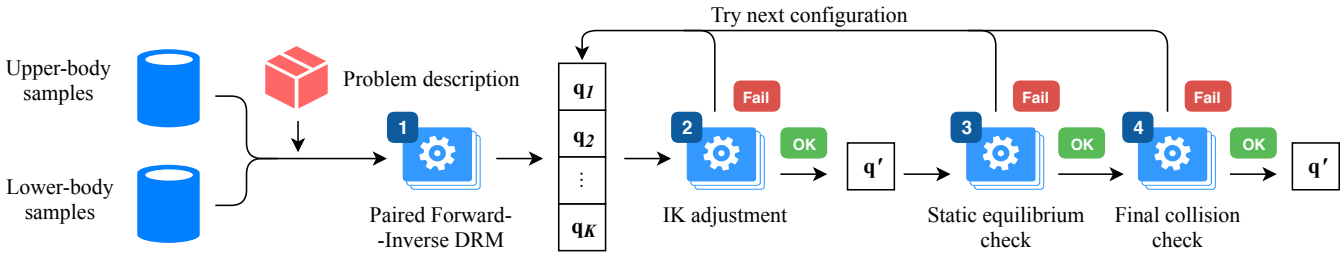


Fig. 3: Overview of the proposed planning pipeline. The numbered blocks with a gear represent the key *stages* along the pipeline.

where $R(\mathbf{q})$ is the static equilibrium robustness of the whole-body configuration $\mathbf{q}$ (i.e., b_0 in optimization problem 1), w_T is a weight for the distance between a configuration's grasping tip $T_{rhand}(\mathbf{q})$ and the target pose $T_{rhand}(\mathbf{q}^*)$, and w_R is another weight for the robustness measure of the static equilibrium of configuration $\mathbf{q}$. Once sorted, candidate configurations are tested in order. Planning terminates when a whole-body candidate successfully reaches the end of the pipeline, or the candidate set is exhausted.

Stage 2 consists of an Inverse Kinematics (IK) adjustment to ensure the target reaching constraints are satisfied and that the feet are in perfect contact with the support regions.[3] Because the robust static equilibrium constraint is not part of the IK formulation problem, the adjusted configuration, $\mathbf{q}'$, must go through *Stage 3*, where its static equilibrium robustness is computed. Finally, a full collision check has to be performed over $\mathbf{q}'$ in *Stage 4*. The reason for this being that, if the IK adjustment involves a considerable kinematic displacement, the occupancy encoding of candidate $\mathbf{q}$ (which lead to $\mathbf{q}'$) might no longer hold. Stages 2, 3 and 4 are repeated for each candidate configuration selected in *Stage 1* until one of them successfully passes all the stages.

Figure 4 shows task snapshots of the results obtained after embedding our proposed planning pipeline into our higher-level control framework. Given a bimanual grasping target, the previously described pipeline finds a valid stance location and a reachable whole-body configuration. Afterwards, the feet locations of this result are passed on to a footstep planner [3] to generate a walking trajectory, bringing the robot to the desired stance. Finally, after having arrived at the computed standpoint, a whole-body motion planner [4] is invoked to generate a collision-free whole-body motion to reach the desired pre-grasping configuration.

V. Evaluation

Based on our previous research and on the trade-off between memory consumption and mapping completeness, we have chosen to discretize the workspace into $10\,\mathrm{cm}$ voxels. Furthermore, at *Stage 1*, a whole-body configuration, $\mathbf{q}$, must respect the following constraints in order to be admitted as a candidate: (i) the z-distance between each foot and its support region must be less than half the discretized workspace resolution — $5\,\mathrm{cm}$ in this case; (ii) the orientation difference between each foot and its support region, i.e., the angle measured between the normal vector of the support region and the foot's normal, must be less than a certain tolerance — we used $0.25\,\mathrm{rad}$. At *Stage 2*, we set the following task-specific tolerances for satisfying the equality constraints: $1\,\mathrm{mm}$ for hands and feet positions; $10^{-5}\,\mathrm{rad}$ for hand orientations; and $10^{-3}\,\mathrm{rad}$ for feet orientation. At *Stage 3*, we use four generators per contact and the contact friction coefficient μ is set to 0.3 — which is comparable to half the friction between rubber and dry concrete on clean and dry surfaces. These are the same parameters used for computing the lower-body dataset. The work presented in this paper was implemented using EXOTica [15] and all evaluations carried out in a single-threaded process on a $4.00\,\mathrm{GHz}$ Intel Core i7-6700K CPU with $32\,\mathrm{GB}$ $2133\,\mathrm{MHz}$ RAM.

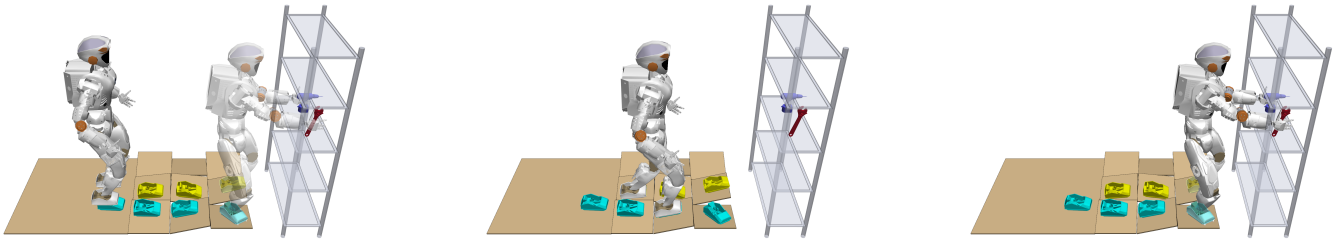
[3]Here, we use the optimization-based IK from Drake [14].

TABLE II: Upper-body map construction analysis.

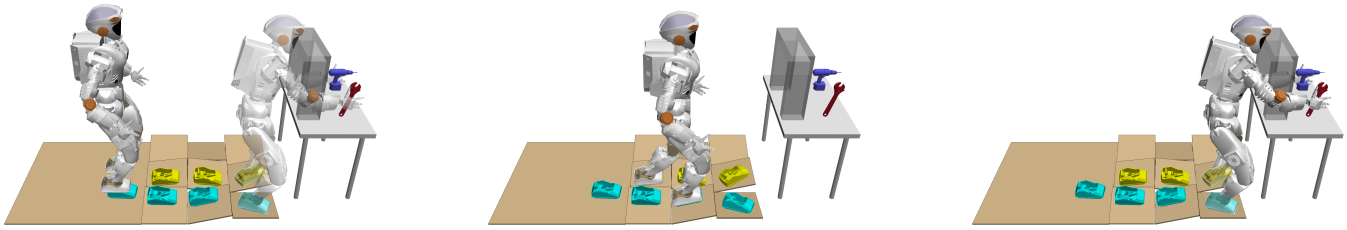
Designation	Constrained	No. of samples	Build time (hh:mm:ss)	Size (MB)
Υ_{500K_c}	Yes	5×10^5	02:54:35	645
Υ_{1M_c}	Yes	1×10^6	05:39:21	1 290
Υ_{2M_u}	No	2×10^6	04:20:23	2 183
Υ_{2M_c}	Yes		10:45:49	2 186
Υ_{4M_u}	No	4×10^6	08:40:46	4 366
Υ_{4M_c}	Yes		21:34:39	4 372

Table II shows the details of the upper-body datasets we have generated for this work. An upper-body dataset is designated by the symbol Υ followed by the number of samples it contains and its type (c - constrained, u - unconstrained) in subscript. The actual size of datasets vary for different robot models as the majority of space is used to encode workspace occupancy. The sizes shown in the table correspond to the model of the 38-DoF NASA Valkyrie humanoid robot. It is clear that both the time it takes to build the dataset and the size it occupies in disk grows with the number of samples. Moreover, it is noticeable that, due to the nature of the rejection sampling process, *constrained* dataset variants take longer to be built than the *unconstrained* ones.

Table III shows the details of the lower-body datasets we have generated for this study. A lower-body dataset is designated by the symbol Π followed by the number of samples it contains in subscript. Datasets Π_{11K} and Π_{151K} have been generated in a deterministic fashion, by incrementally stepping through joint values and maintaining the feet horizontal. All datasets other than Π_{11K} and Π_{151K} have been generated using our new lower-body sampling



(a) Grasping a drill and a wrench in a shelf compartment on complex terrain with multiple support regions at different inclinations.



(b) Grasping a drill and wrench similarly to the previous task but through a narrow frame atop a table. This scenario is purposefully built in such a way that the set of feasible solutions is comprised *only* of whole-body configurations where the left arm of the robot passes through the frame on top of the table.

Fig. 4: Snapshots of task stage progression in time on two scenarios with the same terrain but different surroundings. The robot needs to reach for a drill and a wrench in a shelf compartment and atop a table, respectively. The spatial location of the targets are the same in both scenarios. Without change, our method automatically adapts to the different surroundings and returns suitable whole-body configurations.

TABLE III: Lower-body map construction analysis.

Designation	R_{min}	No. of samples	Build time (hh:mm:ss)	Size (MB)
Π_{1K}	10	1 000	00:07:14	1
Π_{10K_1}	0	10 000	00:30:53	7
Π_{10K_2}	10		01:11:24	7
Π_{10K_3}	20		04:27:00	8
Π_{50K_1}	0	50 000	02:38:17	38
Π_{50K_2}	10		05:59:24	37
Π_{11K}	-	11 812	00:01:45	7
Π_{151K}	-	151 503	00:22:17	94

strategy. A correlation between the time taken and the strategy used to create a dataset can be observed: the proposed rejection sampling method takes longer as large quantities of low quality samples are rejected. Moreover, the higher the threshold for R_{min}, the longer the dataset generation time. However, as sampling takes place offline, dataset build times can often be neglected.

Figure 5 shows the testbed we have modeled in order to carry out evaluation tests of our planning method. It is comprised of eighteen support regions with different inclinations. The layout is reconfigurable, enabling us to change the arrangement of the support regions if necessary.

A. Obstacle-Free Benchmark

We created a benchmark to evaluate the performance of our method in obstacle-free environments on the aforementioned testbed (cf. Figure 5). The benchmark tests a total of 1000 planning requests. Each request is generated by sampling a pseudo-random upper-body configuration using the same constraints as during dataset creation. Afterwards, a random

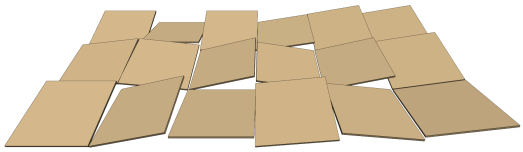


Fig. 5: Testbed of inclined support regions used during the benchmarking tests of our planning method. The testbed is organized in three rows, each with six supports. Each support is $40\,\mathrm{cm} \times 40\,\mathrm{cm}$ and has an inclination of either $10°$ or $15°$.

yaw is applied to the configuration. Finally, the pose is translated to a random location in the testbed, subject to the condition of its pelvis projection lying within the shrunk x-y boundaries of the testbed.

Table IV shows the benchmark results of different lower-body datasets while using the same constrained upper-body dataset, Υ_{2M_c}. The two bottom rows of the table (i.e., Π_{11K} and Π_{151K}) concern lower-body datasets generated according to the methodology of our previous work. The remaining rows concern lower-body datasets generated with our most recent approach. The benchmark results show that our sampling strategy significantly outperforms the previous, which is reflected under the request success rate column. The number of "Total Candidates" shows how well the dataset matches the request criteria during the benchmark. The smallest dataset tested, Π_{1K}, produces very few candidates, which results in short computation times but also reduces planning success rate. Moreover, the quality of the candidates is reflected in the number of "Rejected Candidates", which indicates how many candidate solutions were discarded further down the pipeline until an actual feasible solution was found. The rejection rates of the datasets generated using the proposed method (i.e., Π_{10K_i} and Π_{50K_i}) are an order of magnitude lower than the ones generated in our previous work (i.e., Π_{11K} and Π_{151K}). Finally, the results also show that the minimum

TABLE IV: Comparison of different lower-body sampling methods using a constrained upper-body dataset, Υ_{2M_c}.

Dataset	No. of samples	R_{min}	Success rate (%)		Candidates		Durations (ms)		
			Stage 1	Request	Total	Rejected	UB filter	IK	Request
Π_{1K}	1 000	10	82.9	72.5	9.4 ± 20.5	0.69 ± 1.38	87 ± 17	18.8 ± 2.3	191 ± 77
Π_{10K_1}	10 000	0	97.5	**94.0**	106.5 ± 304.1	2.01 ± 3.79	90 ± 18	18.2 ± 1.9	436 ± 303
Π_{10K_2}		10	97.3	**93.8**	116.1 ± 344.3	1.91 ± 4.45	89 ± 19	18.4 ± 2.6	457 ± 369
Π_{10K_3}		20	96.9	**93.4**	113.9 ± 245.6	1.75 ± 4.05	86 ± 17	18.0 ± 2.2	421 ± 268
Π_{50K_1}	50 000	0	99.4	**98.6**	441.5 ± 1017.7	2.01 ± 3.74	88 ± 17	18.1 ± 3.4	1094 ± 898
Π_{50K_2}		10	99.3	**98.9**	567.3 ± 1874.2	2.37 ± 4.04	87 ± 18	18.0 ± 2.3	1244 ± 1528
Π_{11K}	11 812	-	26.1	23.8	544.7 ± 1690.1	19.27 ± 118.30	84 ± 17	18.1 ± 2.9	1370 ± 3948
Π_{151K}	151 503	-	28.6	27.4	$7300.8 \pm 25\,004.4$	108.86 ± 1198.77	85 ± 16	17.5 ± 1.5	$9344 \pm 29\,810$

TABLE V: Detailed analysis of benchmark in the "shelf" environment.

Upper-body	Lower-body	Success rate (%)		Candidates		Durations (ms)		
		Stage 1	Request	Total	Rejected	UB filter	IK	Request
Υ_{2M_c}	Π_{10K_2}	64.1	57.2	16.6 ± 65.7	0.37 ± 0.93	95 ± 7	17.8 ± 1.5	195 ± 98
	Π_{50K_2}	77.5	74.1	105.0 ± 547.4	0.63 ± 1.28	94 ± 7	17.4 ± 1.4	299 ± 427
Υ_{4M_c}	Π_{10K_2}	68.9	63.0	22.3 ± 123.3	0.41 ± 1.72	166 ± 15	18.1 ± 1.9	318 ± 168
	Π_{50K_2}	84.9	**81.8**	104.9 ± 564.3	0.73 ± 2.21	152 ± 12	17.4 ± 1.7	382 ± 445

robustness threshold R_{min} does not affect the overall request success rate. However, the robustness threshold does affect the resilience of the system to external disturbances. Yet, this was not evaluated in this benchmark.

TABLE VI: End-pose planning failure analysis using a constrained upper-body dataset.

Name	Failure stage decomposition (%): Static Equilibrium Check (SEC), and Final Collision Check (FCC)
Π_{1K}	SEC: 58.0% / FCC: 41.5%
Π_{10K_1}	SEC: 79.3% / 20.7%
Π_{10K_2}	SEC: 74.2% / FCC: 25.7%
Π_{10K_3}	SEC: 70.8% / FCC: 29.0%
Π_{50K_1}	SEC: 78.6% / 20.6%
Π_{50K_2}	SEC: 75.1% / FCC: 24.8%

Table VI provides some valuable insight regarding the pipeline stages at which a test candidate got rejected. That is, from all the "Rejected Candidates" in Table IV, Table VI breaks down at which point in the pipeline a candidate was rejected. With this information we can understand why the candidates failed and which is the most predominant factor for rejecting a candidate. Each entry contains a color bar with, from left to right, the percentage of test-candidates rejection during IK adjustment (*Stage 2*, purple), static equilibrium check (SEC) (*Stage 3*, pale blue), and final collision check (FCC) (*Stage 4*, dark blue). Failure due to the IK adjustment is minimal ($0.8\,\%$, in the Π_{50K_1} entry, is the greatest percentage in the tests we carried out). The most frequent stage at which a sample gets rejected is the SEC. Furthermore, a correlation exists between the minimum robustness threshold, R_{min}, and the SEC-FCC rejection distribution: as the robustness storage criteria gets more demanding, rejection starts to shift from occurring during the SEC to the FCC stage (cf. Π_{10K_1}, Π_{10K_2}, and Π_{10K_3}). The conclusion to draw from this table is that the bottleneck is due to the adjustment performed by the IK solver during *Stage 2*, which returns a whole-body configuration that is no longer in robust static equilibrium.

B. Shelf Benchmark

We created a second benchmarking test to evaluate the performance of our method in environments cluttered with obstacles. For that, we make use of the same testbed terrain of the previous benchmark, with the addition of a shelf. The benchmark routine translates the shelf in incremental steps about the testbed, three steps in the y-axis direction, and 21 steps in the x-axis direction. Finally, for each position of the shelf, 16 bimanual requests located inside one of the shelf compartments are passed on as inputs to our planning pipeline. This amounts to a total of 1008 requests during a whole benchmark session.

Table V shows a detailed analysis of the benchmark results. The meaning of data presented under each column follows the same convention as in Table IV — please confer V-A for a detailed description of what is listed under each column.

By inspecting Table V we can observe that increasing the number of samples in the lower-body dataset (e.g., using Π_{50K_2} instead of Π_{10K_2}) significantly increases the total number of candidates per request. This is ideal, since a wider set of available whole-body configurations translates into more variability, which in turn increases the chances of successfully finding a solution when dealing with cluttered environments — cf. the percentage increase in the "Request, Success rate" column. The downside of having more options available is the time increase required to process and choose the best

TABLE VII: Benchmark request *success rate* and *duration* analysis for varying dataset sizes. Each cell contains two lines: (1) results concerning the "obstacle-free" environment, and (2) results obtained for the "shelf" environment.

Lower-Body	Upper-body			
	Υ_{500K_c} (500K)	Υ_{1M_c} (1M)	Υ_{2M_c} (2M)	Υ_{4M_c} (4M)
Π_{1K} (1K)	43.5% / 78 ± 54 ms	55.7% / 133 ± 77 ms	72.5% / 191 ± 77 ms	80.6% / 355 ± 138 ms
	7.3% / 71 ± 33 ms	11.8% / 94 ± 41 ms	24.8% / 153 ± 60 ms	25.7% / 222 ± 71 ms
Π_{10K_2} (10K)	81.7% / 184 ± 121 ms	90.3% / 271 ± 214 ms	93.8% / 457 ± 369 ms	97.7% / 733 ± 507 ms
	34.3% / 104 ± 74 ms	38.2% / 122 ± 65 ms	57.2% / 195 ± 98 ms	63.0% / 318 ± 168 ms
Π_{50K_2} (50K)	92.9% / 428 ± 655 ms	96.0% / 787 ± 756 ms	98.9% / 1244 ± 1528 ms	99.5% / 2294 ± 2446 ms
	52.3% / 138 ± 121 ms	47.2% / 159 ± 142 ms	74.1% / 299 ± 427 ms	81.8% / 382 ± 445 ms

solution. This downside can be prevented by employing a good cost function to sort candidate poses. The "Rejected Candidates" column shows that, on average, fewer than one candidate had to be re-tested beyond the initial candidate until a feasible solution was found. In other words, for most of the end-pose planning requests, the candidate on the top of the sorted candidates list was indeed a valid (and the chosen) solution. Thus, providing enough evidence to support that the employed cost function is reliable and adequate.

C. Obstacle-Free vs. Shelf Benchmark Remarks

Table VII presents a comparison between the results of the obstacle-free and shelf benchmarks for upper- and lower-body datasets of different sizes. Results show that dataset size has a greater impact on success rates for cluttered environments rather than obstacle-free environments. Note that the planning request durations tend to be much longer for the obstacle-free environment when compared to the shelf environment. This is due to fewer configurations being invalidated by the configuration-to-workspace-occupancy encoding and thus more candidates having to be scored and ranked.

VI. Conclusions

This paper presents a method for whole-body end-pose planning on inclined supports in complex environments taking contact properties such as slope and friction into account. This work also analyzes the impact of including static equilibrium robustness as part of the sampling heuristics for the offline preprocessing stage to improve dataset quality leading to reduced planning times and increased algorithm success rates while using smaller datasets and covering larger state spaces. In particular, offloading the computation of static stability properties to the preprocessing stage allows the algorithm to propose candidate poses with higher quality and chance of success. We have validated our approach with the 38-DoF NASA Valkyrie humanoid in full physics simulation showing that the proposed method results in physically achievable, robust configurations on inclined surfaces.

Splitting the kinematic structure of non-homogeneous legged robots leads to a greater coverage of the $\mathcal{C}$-space through the recombination step, however, also a vast number of possible candidate poses (which grows exponentially with dataset size) that need to be checked and ranked, which might potentially slow down the planning process. In this work we addressed this challenge through an informed and effective cost function, leaving further speed-ups through parallelization to future work. An analysis of the different failure stages during an extensive benchmark shows the main bottleneck is due to adjusted samples no longer meeting the static equilibrium criteria. A possible step to addressing this issue is to explicitly include the robustness measure in the formulation of the IK optimization problem (*Stage 2*). However, finding a differentiable proxy metric is non-trivial and an interesting avenue for future work. Finally, it would be desirable to plan footsteps in a continuous fashion, akin to [16], in order to overcome the uncertainty of accumulated state estimation drift during locomotion, and to actively re-plan walking trajectories to avoid dynamic obstacles.

References

[1] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettré, "A reachability-based planner for sequences of acyclic contacts in cluttered environments," in *Robotics Research*. Springer, 2018.

[2] Y. Yang, V. Ivan, Z. Li, M. Fallon, and S. Vijayakumar, "iDRM: Humanoid motion planning with realtime end-pose selection in complex environments," in *IEEE Humanoids*, 2016.

[3] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *IEEE Humanoids*, 2014.

[4] Y. Yang, V. Ivan, W. Merkt, and S. Vijayakumar, "Scaling sampling-based motion planning to humanoid robots," in *IEEE ROBIO*, 2016.

[5] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai *et al.*, "An architecture for online affordance-based perception and whole-body planning," *Journal of Field Robotics*, 2015.

[6] F. Burget and M. Bennewitz, "Stance selection for humanoid grasping tasks by inverse reachability maps," in *IEEE ICRA*, 2015.

[7] Y. Yang, W. Merkt, H. Ferrolho, V. Ivan, and S. Vijayakumar, "Efficient Humanoid Motion Planning on Uneven Terrain Using Paired Forward-Inverse Dynamic Reachability Maps," *IEEE RA-L*, 2017.

[8] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *IEEE IROS*, 2007.

[9] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *IEEE ICRA*, 2013.

[10] T. Bretl and S. Lall, "Testing static equilibrium for legged robots," *IEEE Transactions on Robotics*, 2008.

[11] S. Caron, Q.-C. Pham, and Y. Nakamura, "Leveraging Cone Double Description for Multi-contact Stability of Humanoids with Applications to Statics and Dynamics," in *Robotics: Science and System*, 2015.

[12] S. Barthélemy and P. Bidaud, *Stability measure of postural dynamic equilibrium based on residual radius*. Springer, 2008.

[13] A. Del Prete, S. Tonneau, and N. Mansard, "Fast algorithms to test robust static equilibrium for legged robots," in *IEEE ICRA*, 2016.

[14] R. Tedrake and the Drake Development Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: https://drake.mit.edu

[15] V. Ivan, Y. Yang, W. Merkt, M. P. Camilleri, and S. Vijayakumar, *EXOTica: An Extensible Optimization Toolset for Prototyping and Benchmarking Motion Planning and Control*. Springer, 2019.

[16] M. F. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, "Continuous humanoid locomotion over uneven terrain using stereo fusion," in *IEEE Humanoids*, 2015.